Cover Song Identification in Practice with Multimodal Co-Training

Simon Hachmeier, Robert Jäschke

L3S Research Center, Appelstr. 9a, Hanover, 30167, Germany
School of Library and Information Science, Dorotheenstr. 26, Humboldt-Universität zu Berlin, Berlin, 10117, Germany

Abstract
The task of cover song identification (CSI) deals with the automatic matching of audio recordings by modeling musical similarity. CSI is of high relevance in the context of applications such as copyright infringement detection on online video platforms. Since online videos include metadata (eg. video titles, descriptions), one could leverage it for more effective CSI in practice. In this work, we experiment with state-of-the-art models of CSI and entity matching in a Co-Training ensemble. Our results outline slight improvements of the entity matching model. We further outline some suggestions for improvements of our approach to overcome the issue of overfitting CSI models which we observed.

Keywords
co-training, cover song identification, entity matching

1. Introduction

Cover song identification (CSI) aims at matching audio recordings to their respective musical cliques based on musical similarity. One typical application of CSI is copyright infringement detection on online video platforms or social networks.

Recent state-of-the-art CSI models have shown great success [1, 2, 3, 4, 5, 6, 7]. However, these models are solely audio-based. Prior approaches have also demonstrated the effectiveness of metadata for the task [8, 9].

In this work, we model the task of CSI as a multimodal problem incorporating music similarity and entity matching. We design a Co-Training algorithm that leverages the natural split of two views: a text view and an audio view. We utilize the two models to iteratively generate pseudo labels for each other for an unlabeled dataset of YouTube videos. We evaluate the performance of both models on publicly available CSI datasets. In the following, we first introduce into Co-Training and outline some related work. We then propose our Co-Training algorithm, and document details about our dataset and implementation in Section 3 to Section 5. In our experiments in Section 6 we show results before closing this paper with Section 7 outlining some ideas to improve our approach.

LWDA'23: Lernen, Wissen, Daten, Analysen. October 09–11, 2023, Marburg, Germany
hachmeier@l3s.de (S. Hachmeier); jaeschke@l3s.de (R. Jäschke)

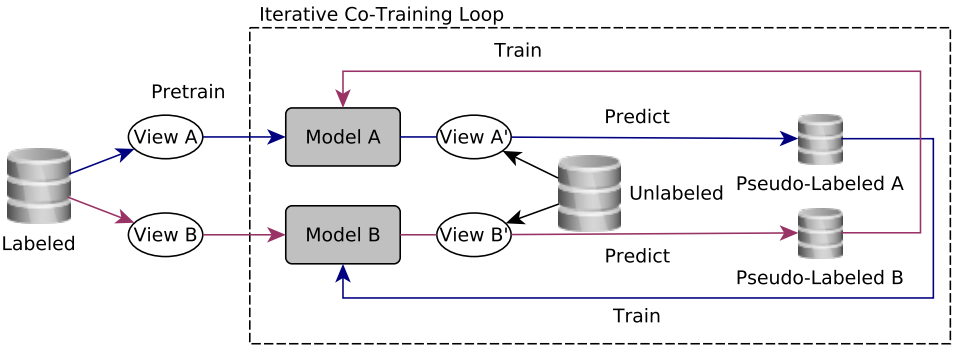


Figure 1: Own illustration of Co-Training with two views and models.

2. Preliminaries and Related Work

Co-Training was initially proposed by Blum and Mitchell [10] and refers to the idea to leverage automatically generated pseudo labels to improve the performance of models in the ensemble. This enables training models in cases where only a small subset of the available data is labeled, which applies to many real world scenarios. Co-Training relies on the availability of multiple views which are required to fulfill the following assumptions:

1. *Sufficiency*: Each view is sufficient to address the task at hand.
2. *Independence*: The views are conditionally independent.

As illustrated in Figure 1, the models within the ensemble iteratively provide pseudo labels for each other. One key component is the selection of a fraction of predictions as pseudo labels, based on a constraint such as confidence thresholds [11, 12], limiting to a fraction of most confident samples by ranking [13, 14] or other methods [15, 16, 17, 18].

Recently, various applications of Co-Training with deep learning models have been proposed for the modalities text [13, 16], images [14, 19, 11, 12, 20, 21] and on multiple modalities [22, 15]. Lang et al. [13] improve prompt based learning by using GPT-3 output probabilities and frozen representations of openly available large language models to improve prompt-based learning. Both of their proposed methods for pseudo label selection are based on the ranking of samples. An approach by Wu et al. [16] applies Q-learning to improve the selection policy for the partition of unlabeled data to be pseudo-labeled. They demonstrate the effectiveness of reinforced Co-Training on text classification tasks. Peng et al. [14] use adversarial examples in an ensemble of multiple models for model diversity to improve the ensemble performance for image segmentation on medical images. To select pseudo labels, a fraction parameter is used that increases over iterations. In contrast, Yang et al. [11] apply a Co-Training framework with a fixed threshold to the task of domain adaption task. Xian and Hu [12] use a fixed threshold parameter for pseudo-labeling in the task of person re-identification.

Some approaches successfully make use of views arising from modalities. A multimodal approach of Hinami et al. [15] leverages multiple views of the modalities text, audio and video

found in web videos to improve concept classification. Pseudo labels are selected based on a voting approach within the ensemble. Another multimodal approach with text and images from web articles of Bhattacharjee et al. [22] improves the task of fake news detection. Their pipeline includes an attention-aware step to fuse two views. The models then co-train based on sampling of hard positive samples. In the following, we present our Co-Training algorithm which is based on fixed-thresholds.

3. Multimodal Co-Training Algorithm

We have access to an entity matching model TM based on a language model and a cover song identification model AM based on metric learning. Both models are pretrained and achieve state-of-the-art performance for the task at hand. However, we aim to improve their performance by training these models on a labeled dataset D_L and an unlabeled dataset D_U. Each item v within either of the datasets is a YouTube video representation v which is represented by a text view (YouTube metadata) and an audio view based on audio features (cf. Section 5).

Accordingly, $TM(v_i, v_j)$ computes the entity matching confidence $0 < tm < 1$ for a pair $v_i, v_j \in D_U \cup D_L$ and $AM(v_i, v_j)$ the musical similarity modeled as cosine similarity $-1 < am < 1$. A labeled item $v \in D_L$ has a known clique or musical work where it belongs to represented by $w(v) \in W$. Unlabeled items $v \in D_U$ have a candidate clique $\hat{w}(v) \in W$. It is unknown whether v belongs to this clique. However, among all possible cliques this is the most likely one, because v was found with queries formulated to find items for this clique as explained in Section 4.

We argue that we can address the problem of multimodal CSI by Co-Training. Since both models are pretrained, we expect the first Co-Training assumption (cf. Section 2) to hold. We further argue that both views are conditionally independent, due to the natural split given by modalities.

In Algorithm 1, we show the Co-Training loop. We randomly sample three labeled ($l = 3$) and three unlabeled videos ($u = 3$) for two randomly selected cliques ($s = 2$) per iteration. We make use of hard threshold parameters. The output of TM is based on softmax layers on top of a language model. We therefore simply denote γ as the outer boundary for confidence, indicating that a pseudo label is either positive if $tm > 1 - \gamma$ or negative if $tm < \gamma$. For the audio model we impose two thresholds. We observed that the CSI model we use does not output equally distributed similarity values spreading to the boundaries of the cosine similarity. Hence, we use a lower threshold to set negative pseudo labels if $am < \tau_{\text{LOWER}}$ and an upper threshold to set positive pseudo labels if $am < \tau_{\text{UPPER}}$.

In Algorithm 2, we show one iteration of Co-Training where $\text{MAX}(M, 0)$ denotes the element-wise maximum operator applied to the matrix M and 0 and $\overline{\text{MAX}}(M)$ and denotes the respective row-wise maximum operation applied to M.

We first predict the entity matching confidences and musical similarities for each pair of items within all the pairs in the batch and assign those to matrices $\hat{Y}_{\text{text}}$ and $\hat{Y}_{\text{audio}}$. Subsequently, the similarity square matrices are masked to retain only pairwise relationships with a known ground truth label from D_L or with a sufficiently confident pseudo label. As masking values, we select 1 to represent a indicating a positive relationship among the items (both are from the

Algorithm 1 Multimodal Cover Song Co-Training Loop

Initialize
Maximum number of iterations I_{max}, Number of cliques per batch s, set of clique identifiers W, number of labeled items per batch l, number of unlabeled items per batch u, outer boundary for text model γ, lower threshold for audio model τ_{LOWER}, upper threshold for audio model τ_{UPPER}, labeled dataset D_L, unlabeled dataset D_U, learning rate η, audio model AM, text model TM

for $i \longleftarrow$ to I_{max} **do**

    Sample $W_B = \{w_1, \ldots, w_s\}$ from W ´

    for $w \in W_B$ **do**:
        Sample $L^w = \{v_1, \ldots, v_q\}$ from D_L where $w(v) \in W_B$
        Sample $U^w = \{\hat{v}_1, \ldots, \hat{v}_c\}$ from D_U where $\hat{w}(\hat{v}) \in W_B$
    end for

    $L_B = \bigcup_{w \in W_B} L^w$
    $U_B = \bigcup_{w \in W_B} U^w$

    CoTrainIter$(L_B, U_B, \tau_{\text{LOWER}}, \tau_{\text{UPPER}}, \gamma, \eta, AM, TM)$

end for

same clique) and -1 to indicate the contrary. Additionally, we select 0 to mask out uncertain relationships for pairs without a ground truth label and insufficient confidence of the model generating the pseudo label. The pseudo label masks M_{text} and M_{audio} are used to sample the similarity values to use for training updates with hard triplet mining as proposed by Xuan et al. [23] and applied to train prior CSI models [24, 25]. The lowest distances of the positive relationships in DIST^{+}_{audio} and DIST^{+}_{text} and the highest distances for the pairwise negative relationships DIST^{-}_{audio} and DIST^{-}_{text} represent the components of the hard triplets that are used for the training updates.

We train the metric learning model AM with triplet loss which is defined as:

$$L_i^{\text{tri}} = \max(D(v_i, v_+) - D(v_i, v_-) + m, 0), \tag{1}$$

where $m = 1$ is the margin parameter, v_+ and v_- are the positive and negative to anchor v_i which are used to compute the distances $D(v_i, v_+)$ and $D(v_i, v_-)$ as found in DIST^{+}_{audio} and DIST^{-}_{audio} respectively.

Our entity matching model TM is based on a large language model which we train with binary cross entropy loss:

Algorithm 2 Co-Training Iteration for One Batch (Triplet Loss with Hard Triplet Mining).

Initialize

Set of labeled items per batch L_B, set of unlabeled items per batch U_B, outer boundary for text model γ, lower threshold for audio model τ_{LOWER}, upper threshold for audio model τ_{UPPER}, learning rate η, audio model AM, text model TM

Set $n = |L_B \cup U_B|$

Predict

Init. empty matrix $\hat{Y}_{audio} \in \mathbb{R}^{n \times n}$

$\hat{Y}_{audio}[i,j] = AM(v_i, v_j)$ where $v_i, v_j \in L_B \cup U_B$

Init. empty matrix $\hat{Y}_{text} \in \mathbb{R}^{n \times n}$

$\hat{Y}_{text}[i,j] = \text{TM}(v_i, v_j)$ where $v_i, v_j \in L_B \cup U_B$

Ground Truth Square Mask

Init. empty matrix $M_{label} \in \mathbb{R}^{n \times n}$

$$M_{label}[i,j] = \begin{cases} 1 & \text{if } w(v_i) = w(v_j) \\ -1 & \text{if } w(v_i) \neq w(v_j) \\ 0 & \text{if } \textbf{undefined} \in \{w(v_i), w(v_j)\} \end{cases}$$

Pseudo Label Masks

Init. empty matrices $M_{audio} \in \mathbb{R}^{n \times n}$ and $M_{text} \in \mathbb{R}^{n \times n}$

$$M_{audio}[i,j] = \begin{cases} M_{label}[i,j] & \text{if } M_{label}[i,j] \neq 0 \\ 1 & \text{if } \hat{Y}_{audio}[i,j] > \tau_{upper} \\ -1 & \text{if } \hat{Y}_{audio}[i,j] < \tau_{lower} \\ 0, & \text{otherwise} \end{cases}$$

$$M_{text}[i,j] = \begin{cases} M_{label}[i,j] & \text{if } M_{label}[i,j] \neq 0 \\ 1 & \text{if } \hat{Y}_{text}[i,j] > 1 - \gamma \\ -1 & \text{if } \hat{Y}_{text}[i,j] < \gamma \\ 0, & \text{otherwise} \end{cases}$$

Hard Triplet Mining

$\text{DIST}^{+}_{audio} = 1 - \overline{\text{MIN}}(\text{MAX}(M_{text}, 0) * \hat{Y}_{audio}) \in \mathbb{R}^{n \times 1}$

$\text{DIST}^{-}_{audio} = 1 - \overline{\text{MAX}}(\text{MAX}(-1 * M_{text}, 0) * \hat{Y}_{audio}) \in \mathbb{R}^{n \times 1}$

$\text{DIST}^{+}_{text} = 1 - \overline{\text{MIN}}(\text{MAX}(M_{audio}, 0) * \hat{Y}_{text}) \in \mathbb{R}^{n \times 1}$

$\text{DIST}^{-}_{text} = 1 - \overline{\text{MAX}}(\text{MAX}(-1 * M_{audio}, 0) * \hat{Y}_{text}) \in \mathbb{R}^{n \times 1}$

Loss Computation

$\text{LOSS}_{audio} = L^{\text{tri}}(\text{DIST}^{+}_{audio}, \text{DIST}^{-}_{audio})$

$\text{LOSS}_{text} = L^{\text{ce}}(\text{DIST}^{+}_{text}, \text{DIST}^{-}_{text})$

Update

$\theta_{AM} \leftarrow \theta_{AM} - \eta \Delta \text{LOSS}_{audio}(\theta_{AM})$

$\theta_{TM} \leftarrow \theta_{TM} - \eta \Delta \text{LOSS}_{text}(\theta_{TM})$

Table 1
Datasets with numbers of cliques and songs/videos used in our implementation for training, validation, and testing.

Subset	Dataset	Cliques	Items
Training	*Train-YT*	50	50,395
Training	*Train-SHS*	50	1,121
Validation	*Val-SHS*	882	3,172
Test	*Da-Tacos*	2,797	13,707
Test	*Test-SHS*	50	1,259
Test	*Test-YT*	50	628

$$L_i^{\text{ce}} = \sum_{c=1}^{M} y_{i,c} \log(\hat{y}_i), \tag{2}$$

where $\hat{y}_i$ is one prediction as found in in either DIST_{text}^{-} or DIST_{text}^{+} and hence $y_{i,c} \in \{0, 1\}$. In the following, we outline details about our dataset, preprocessing and training implementation.

4. Dataset

We provide an overview of the datasets used in Table 1 and CSV files containing cliques identifiers and YouTube identifiers[1]. The cliques used for implementation rely on two datasets from prior research in CSI: *SHS100K*[2] for training, validation and testing and *Da-Tacos* [26] for testing.

Based on the test subset of *SHS100K* we formulated around 44 text queries per clique to crawl YouTube[3] to find additional songs for these cliques, similarly to our prior work [27]. We split this crawl into two parts with 50 cliques each. One for training composed of *Train-SHS* (labeled dataset D_L) and *Train-YT* (unlabeled dataset D_U) and one for testing which is composed of *Test-SHS* and *Test-YT*. *Test-SHS* is a subset of songs that are represented by YouTube videos in the initial *SHS100K* test subset and *Test-YT* contains other YouTube videos found by the query procedure. We annotated these 628 crawled videos with the help of two students and up to five workers on Mechanical Turk. We only considered labels with full agreement among students and aggregated the worker labels by majority vote.[4]

For validation we use the validation subset of *SHS100K* which we denote by *Val-SHS*.[5] We additionally use the larger *Da-Tacos* dataset for testing.[6]

[1] https://github.com/progsi/datasets_shs_yt_cotraining
[2] cf. https://github.com/NovaFrost/SHS100K provided by Yu et al. [1]
[3] cf. https://pypi.org/project/youtube-search-python/
[4] We report an agreement in Krippendorff's α of 0.43 (workers) and a Cohen's κ of 0.83 (students). While the worker agreement is quite low, measuring the agreement between students and aggregated labels by majority vote for a subset of 210 songs yields a Cohen's κ of 0.81.
[5] 81% were retrievable from YouTube.
[6] The authors of *Da-Tacos* provide CREMA features publicly. However, we needed to extract CQT spectograms of

For each video, we downloaded the MP3 files with a sampling rate of 22,050 Hertz[7] to extract audio features. We extract CREMA[8] features and constant-Q transform features[9] (CQT) with 84 frequency bins.

Furthermore, we retrieved the metadata for each video. To ensure that semantics are preserved independently of the Unicode font, we mapped various Unicode fonts to basic Latin characters using *Unicodedata*[10].

5. Implementation Details

We use the *BERT*-based entity matching model *Ditto* [28] as our text model which is publicly available on Github.[11] *Ditto* requires fine-tuning specifically to the structure of attributes in the entities, in our case YouTube videos. We use the *SHS100K*-Train subset as *Ditto* pretraining dataset, which does not overlap with dataset any of our other datasets shown in Table 1. Following the splits applied by Li et al. [28] we created a training, validation, and test set with a ratio of 3:1:1 with each containing positive and negative pairs of YouTube videos in a 1:4 ratio. We gathered the negative pairs by randomly sampling videos from another randomly selected work. We use only the video titles and channel names as YouTube metadata representations. We additionally experimented with YouTube descriptions but preliminary results showed inferior results (F1 score of 27% against 95%) to the ones achieved by using only video titles and channels. We used all of the proposed data augmentation techniques and the best performing language model (*RoBerta*) as described in [28]. We applied the best model checkpoint evaluated on the test set after 50 epochs for our matching task.

We use two different state-of-the-art CSI models which are publicly available[12]: *CQTNet* [1] and *Re-MOVE* [3]. In both cases, we initialize the pretrained models from the best model checkpoints provided by the authors.

Re-MOVE processes CREMA features which are a variant of pitch class profiles and mainly represent harmonic information. *CQTNet* processes constant-Q transform features (CQT), which are spectograms with a logarithmically spaced frequency axis.

Following the Co-Training approach by Yang et al. [11], we use stochastic gradient descent as optimizer with learning rate 0.01 and momentum $\in \{0, 0.9\}$. We validate the used audio model and *Ditto* every 100 iterations. Since the prediction of a square matrix is expensive with *Ditto*, we initialize a random subset of the validation set at the beginning of each training and use it throughout the training.

MP3s for CQTNet. Hence, we only include the subset of videos which were available on YouTube which makes up around 92% of full *Da-Tacos*.

[7] cf. https://github.com/yt-dlp/yt-dlp

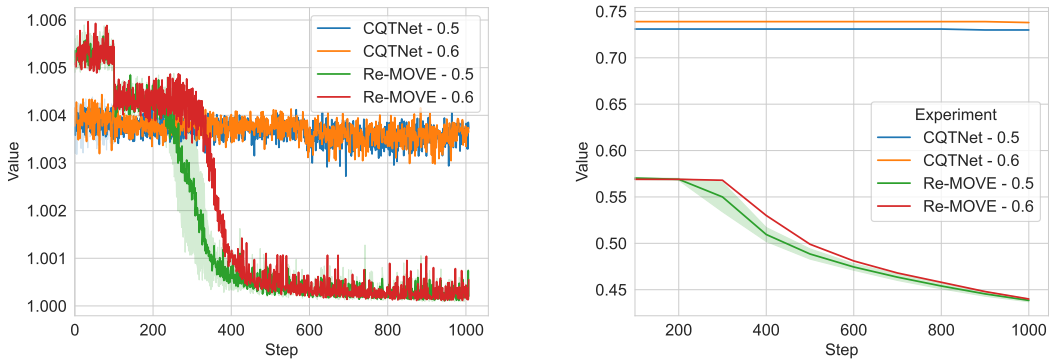
[8] cf. https://github.com/bmcfee/crema

[9] cf. https://librosa.org/doc/latest/index.html

[10] cf. https://docs.python.org/3/library/unicodedata.html

[11] cf. https://github.com/megagonlabs/ditto

[12] We experimented with the *ByteCover* implementation by Orfium: https://github.com/Orfium/bytecover However, the implementation was not provided by the authors of the paper and achieves lower performance than both models we use.



(a) Loss of audio models.

(b) Validation mAP of audio models.

Figure 2: Comparison of Audio Models for first 1,000 iterations and momentum of 0.

6. Experiments

We evaluate our proposed Co-Training algorithm on ensembles with *Ditto* [28] as TM paired with one of the pretrained audio models *CQTNet* [1] and *Re-MOVE* [3] as AM. Our provided baselines are the pretrained models before Co-Training. We further compare to a simple baseline: the Levensthein-based function token set ratio[13]. We report the mean average precision (mAP) which is the main evaluation metric used in cover song identification [1, 2, 3, 4, 5, 6, 7]. Results are shown in Table 2 for the two best ensembles we found per pair of TM and AM:

- *Co-CQT*: with *CQTNet* and $\gamma = 0.1$, $\tau_{\text{UPPER}} = 0.7$, $\tau_{\text{LOWER}} = 0.2$.
- *Co-ReM*: with *Re-MOVE* and $\gamma = 0.2$, $\tau_{\text{UPPER}} = 0.5$, $\tau_{\text{LOWER}} = 0.3$.

6.1. Experiment 1: *CQTNet* Versus *Re-MOVE*

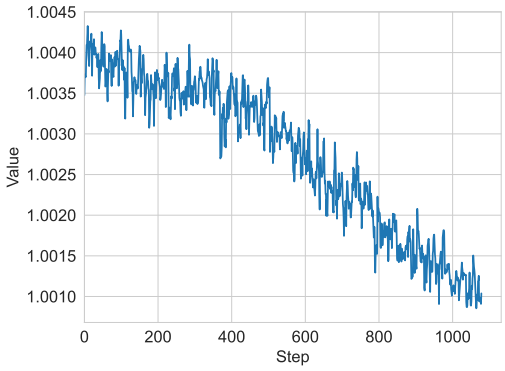
We compare the two audio models with $\tau_{\text{UPPER}} \in \{0.5, 0.6\}$, $\gamma = 0.2$ and $\tau_{\text{LOWER}} = 0.3$. In Figure 2 we show the triplet loss over 1,000 iterations as well as the validation mAP. The strong observable drop in mAP and loss for *Re-MOVE* strongly reflects an overfit. As we show in Table 2, *Re-MOVE* generally performs worse than *CQTNet*. We therefore focus on experimenting with various different thresholds for *CQTNet*. We further observe that the convergence of the loss of *CQTNet* is rather slow. Thus, we impose a momentum of 0.9 in the next experiments.

6.2. Experiment 2: *CQTNet* Threshold Tuning

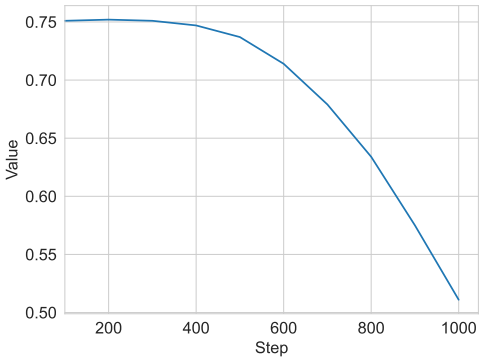
We experimented with different hyperparameter configurations: $\gamma \in \{0.1, 0.2, 0.49\}$, $\tau_{\text{UPPER}} \in \{0.5, 0.6, 0.7\}$, $\tau_{\text{LOWER}} \in \{0.2, 0.3, 0.4\}$.

In Figure 3 we show the loss and validation mAP of *Co-CQT*. We observe that *CQTNet* overfits, shown by the jointly decreasing loss and mAP. The triplet loss converges rather close to the margin for the triplet loss $m = 1$. We observed this result consistently across configurations.

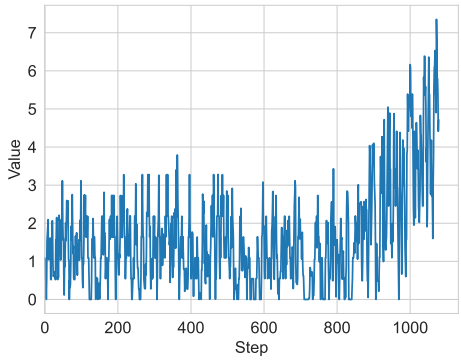
[13] cf. https://github.com/maxbachmann/RapidFuzz



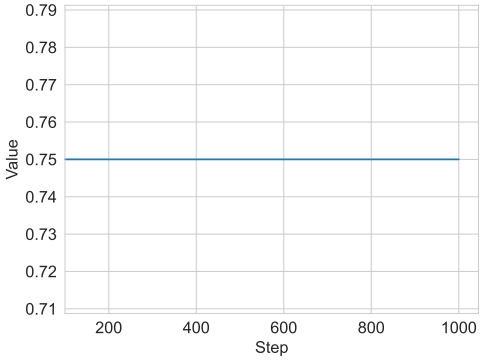
(a) Triplet Loss.



(b) CQTNet Validation mAP.



(c) Binary Cross Entropy Loss.



(d) Ditto mAP.

Figure 3: Losses and validation mAPs *Co-CQT*.

However, we as well observe an increase in loss but a constant validation mAP for *Ditto*[14]. As shown in Table 2, *Ditto* is the only model which actually improves with the Co-Training procedure. Given these two key observations, we hypothesize that balancing the two very different models is a key challenge. In the closing section, we therefore outline some of the potential issues with our approach and ideas to address these.

7. Conclusion and Outlook

In this paper, we applied a Co-Training algorithm for multimodal CSI using an audio-based CSI model along with an entity matching model. We slightly improved the entity matching model *Ditto* for our task. This might suggest that further training iterations can improve *Ditto*. However, both audio-based models seem to overfit quite rapidly.

In the following, we outline some ideas which might have an impact on this problem.

[14]Please note that sampling of a subset of 100 items of the full *Val-SHS* as mentioned in Section 5, can have a major impact on the validation mAP.

Table 2
mAP of the ensembles *Co-CQT* (*Ditto* & *CQTNet*) and *Co-ReM* (*Ditto* & *Re-MOVE*). We report for the best ensembles achieved with our tested hyperparameter configurations. *The computation of predictions in the case of *Ditto* is more time complex than for the CSI models. We therefore report the performance on *Da-Tacos* for a random subset of 1,259 items (size of the *Test-SHS* dataset).

Ensemble	Model	*Val-SHS*	*Test-SHS*	*Test-YT*	*Da-Tacos*
-	*Levenshtein*	0.30	0.50	0.26	0.12
-	*Ditto**	0.62	0.80	0.40	0.24
Co-CQT	*Ditto* (best)	-	**0.84**	**0.44**	**0.28**
-	*Re-MOVE*	0.57	0.69	0.43	0.23
Co-ReM	*Re-MOVE* (best)	0.57	0.70	0.44	0.23
Co-ReM	*Re-MOVE* (last)	0.35	0.46	0.29	0.11
-	*CQTNet*	**0.76**	**0.83**	**0.57**	**0.73**
Co-CQT	*CQTNet* (best)	0.75	0.83	0.56	0.74
Co-CQT	*CQTNet* (last)	0.51	0.55	0.32	0.35

Learning Rates. In comparison, *Ditto* seems to learn rather slow while the audio-based models overfit. We believe that different learning rates for both models could help to prevent this imbalance of model convergence. One potential improvement can be a grid search over different learning rates across the models as proposed by Likhosherstov et al. [29]. Alternatively, one could apply different learning rate schedulers like Yang et al. [11]. Our observations also suggest the potential continuation of the pretraining of *Ditto*, possibly with pseudo labels generated by the audio model. Eventually, Co-Training with both models could be done afterwards to avoid the apparent different starting condition of both models.

Hard Triplet Mining. We sample triplets during training based on the hard triplet mining strategy found in metric learning. In the context of Co-Training, adversarial examples can be used as an alternative [14, 19, 20, 21] which encourage view difference. In contrast, hard triplet mining solely ensures that the most difficult triplets are in the batch are utilized for training.

Losses. Some state-of-the-art CSI models rely on multiloss approaches [5, 6, 7] which combine triplet loss with a softmax loss. While triplet loss encourages intra-class compactness, the latter encourages inter-class discrimination [30]. Thus, our approach might neglect inter-class discrimination. Another alternative to the triplet loss is the utilization of the prototypical triplet loss [31] which considers distances between centroids of positive and negative classes instead of distances to individual samples.

Batch Size. We tested different configurations of thresholds. However, the batch size for labeled and unlabeled items per batch was fixed for all experiments and the number of items for both input datasets was equal. We believe that the increase of unlabeled items per batch in contrast to labeled items could enforce that more interesting items are used during training. That is, due to their containment in our crawl rather than the widely used academic dataset *SHS100K*, which is based on the platform *Secondhandsongs*. The platform itself relies on manual

labour by volunteers subject to policies to determine the boundaries between cover songs whereas our crawl is solely subject to the creative spectrum on YouTube.

Label Confidence Estimation. As outlined in Section 2, other label confidence estimation methods can be applied to Co-Training. In this study, we solely experimented with a threshold-based method. Ranking-based methods or possibly more sophisticated methods could further improve our proposed algorithm.

In future experiments, we plan to test the impact of the factors discussed. We hope that we can find configurations of ensembles which can effectively leverage both views to improve the task of multimodal CSI.

References

[1] Z. Yu, X. Xu, X. Chen, D. Yang, Learning a Representation for Cover Song Identification Using Convolutional Neural Network, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 541–545. doi:10.1109/ICASSP40776.2020.9053839.

[2] Z. Yu, X. Xu, X. Chen, D. Yang, Temporal pyramid pooling convolutional neural network for cover song identification, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, International Joint Conferences on Artificial Intelligence Organization, 2019, pp. 4846–4852. URL: https://doi.org/10.24963/ijcai.2019/673. doi:10.24963/ijcai.2019/673.

[3] J. Serrà Julià, F. Yesiler, E. Gómez Gutiérrez, Less is more: faster and better music version identification with embedding distillation, in: Cumming J, Ha Lee J, McFee B, Schedl M, Devaney J, McKay C, Zagerle E, de Reuse T, editors. Proceedings of the 21st International Society for Music Information Retrieval Conference; 2020 Oct 11-16; Montréal, Canada: ISMIR; 2020. p. 884-92, International Society for Music Information Retrieval (ISMIR), 2020.

[4] F. Yesiler, J. Serrà, E. Gómez, Accurate and scalable version identification using musically-motivated embeddings, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 21–25. doi:10.1109/ICASSP40776.2020.9053793.

[5] X. Du, Z. Yu, B. Zhu, X. Chen, Z. Ma, Bytecover: Cover song identification via multi-loss training, ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2020) 551–555.

[6] X. Du, K. Chen, Z. Wang, B. Zhu, Z. Ma, Bytecover2: Towards dimensionality reduction of latent embedding for efficient cover song identification, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2022, pp. 616–620.

[7] S. Hu, B. Zhang, J. Lu, Y. Jiang, W. Wang, L. Kong, W. Zhao, T. Jiang, WideResNet with Joint Representation Learning and Data Augmentation for Cover Song Identification, in: Proc. Interspeech 2022, 2022, pp. 4187–4191. doi:10.21437/Interspeech.2022-10600.

[8] J. B. L. Smith, M. Hamasaki, M. Goto, Classifying derivative works with search, text,

audio and video features, in: 2017 IEEE International Conference on Multimedia and Expo (ICME), 2017, pp. 1422–1427. doi:10.1109/ICME.2017.8019444.

[9] A. A. Correya, R. Hennequin, M. Arcos, Large-scale cover song detection in digital music libraries using metadata, lyrics and audio features, CoRR abs/1808.10351 (2018). URL: http://arxiv.org/abs/1808.10351. arXiv:1808.10351.

[10] A. Blum, T. Mitchell, Combining labeled and unlabeled data with co-training, Association for Computing Machinery, New York, NY, USA, 1998.

[11] L. Yang, Y. Wang, M. Gao, A. Shrivastava, K. Q. Weinberger, W.-L. Chao, S.-N. Lim, Deep co-training with task decomposition for semi-supervised domain adaptation, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 8906–8916.

[12] Y. Xian, H. Hu, Enhanced multi-dataset transfer learning method for unsupervised person re-identification using co-training strategy, IET Computer Vision 12 (2018) 1219–1227.

[13] H. Lang, M. Agrawal, Y. Kim, D. A. Sontag, Co-training improves prompt-based learning for large language models, in: International Conference on Machine Learning, 2022.

[14] Deep co-training for semi-supervised image segmentation, Pattern Recognition 107 (2020) 107269. doi:https://doi.org/10.1016/j.patcog.2020.107269.

[15] R. Hinami, J. Liang, S. Satoh, A. G. Hauptmann, Multimodal co-training for selecting good examples from webly labeled video, CoRR abs/1804.06057 (2018). URL: http://arxiv.org/abs/1804.06057. arXiv:1804.06057.

[16] J. Wu, L. Li, W. Y. Wang, Reinforced co-training, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 1252–1262. URL: https://aclanthology.org/N18-1113. doi:10.18653/v1/N18-1113.

[17] T. Han, W. Xie, A. Zisserman, Self-supervised co-training for video representation learning, in: Proceedings of the 34th International Conference on Neural Information Processing Systems, NIPS'20, Curran Associates Inc., Red Hook, NY, USA, 2020.

[18] Semi-supervised learning combining co-training with active learning, Expert Systems with Applications 41 (2014) 2372–2378. doi:https://doi.org/10.1016/j.eswa.2013.09.035.

[19] S. Qiao, W. Shen, Z. Zhang, B. Wang, A. Yuille, Deep co-training for semi-supervised image recognition, in: V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss (Eds.), Computer Vision – ECCV 2018, Springer International Publishing, Cham, 2018, pp. 142–159.

[20] H. Xie, C. Fu, X. Zheng, Y. Zheng, C.-W. Sham, X. Wang, Adversarial co-training for semantic segmentation over medical images, Computers in biology and medicine 157 (2023) 106736.

[21] Y. Wang, Y. Zhang, Y. Liu, Z. Lin, J. Tian, C. Zhong, Z. Shi, J. Fan, Z. He, Acn: Adversarial co-training network for brain tumor segmentation with missing modalities, in: Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VII 24, Springer, 2021, pp. 410–420.

[22] S. D. Bhattacharjee, J. Yuan, Multimodal co-training fornbsp;fake news identification using attention-aware fusion, Springer-Verlag, Berlin, Heidelberg, 2021.

[23] H. Xuan, A. Stylianou, X. Liu, R. Pless, Hard negative examples are hard, but useful, in:

A. Vedaldi, H. Bischof, T. Brox, J.-M. Frahm (Eds.), Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020, pp. 126–142.

[24] F. Yesiler, J. Serrà, E. Gómez, Accurate and scalable version identification using musically-motivated embeddings, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 21–25. doi:10.1109/ICASSP40776.2020.9053793.

[25] F. Yesiler, J. Serrà, E. Gómez, Less is more: Faster and better music version identification with embedding distillation, in: International Society for Music Information Retrieval Conference, 2020.

[26] F. Yesiler, C. J. Tralie, A. A. Correya, D. F. Silva, P. Tovstogan, E. Gómez, X. Serra, Da-tacos: A dataset for cover song identification and understanding, in: ISMIR, 2019.

[27] S. Hachmeier, R. Jäschke, H. Saadatdoorabi, Music version retrieval from youtube: How to formulate effective search queries?, in: P. Reuss, V. Eisenstadt, J. M. Schönborn, J. Schäfer (Eds.), Proceedings of the LWDA 2022 Workshops: FGWM, FGKD, and FGDB, Hildesheim (Germany), Oktober 5-7th, 2022, volume 3341 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2022, pp. 213–226. URL: https://ceur-ws.org/Vol-3341/WM-LWDA_2022_CRC_7142.pdf.

[28] Y. Li, J. Li, Y. Suhara, A. Doan, W.-C. Tan, Deep entity matching with pre-trained language models, Proceedings of the VLDB Endowment 14 (2020) 50–60. URL: https://doi.org/10.14778%2F3421424.3421431. doi:10.14778/3421424.3421431.

[29] V. Likhosherstov, A. Arnab, K. Choromanski, M. Lucic, Y. Tay, A. Weller, M. Dehghani, Polyvit: Co-training vision transformers on images, videos and audio, CoRR abs/2111.12993 (2021). URL: https://arxiv.org/abs/2111.12993. arXiv:2111.12993.

[30] A. Taha, Y.-T. Chen, T. Misu, A. Shrivastava, L. Davis, Boosting standard classification architectures through a ranking regularizer, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 758–766.

[31] G. Doras, G. Peeters, A prototypical triplet loss for cover detection, in: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020, pp. 3797–3801. doi:10.1109/ICASSP40776.2020.9054619.